LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-569332-REV-1

Extending MC-SURE to Denoise Sensor Data Streams

M. Ndoye, C. Kamath

November 20, 2012

Asilomar Conference on Signals, Systems, and Computers
Monterey, CA, United States
November 4, 2012 through November 7, 2012

Extending MC-SURE to Denoise Sensor Data Streams

Mandoye Ndoye and Chandrika Kamath
Lawrence Livermore National Laboratory
Livermore, California 94550
Email: ndoye1,kamath2@llnl.gov

***Abstract*—We propose a method to adaptively denoise sensor data streams corrupted by noise that can be approximated as additive white Gaussian. This on-line filtering method is based on the Monte-Carlo Stein's Unbiased Risk Estimate (MC-SURE) algorithm, which enables a blind optimization of the denoising parameters for a wide class of filters. We first identify the challenges that arise as the MC-SURE algorithm is adapted to on-line data processing. We then propose a framework to address these challenges and demonstrate the application of the algorithm using real-world datasets.**

I. INTRODUCTION

The increasing affordability of sensors is enabling cost-effective and real (or near-real) time monitoring of complex phenomena and systems such as fusion in tokamak reactors, electric power grids, or large-scale infrastructure networks. The measurements obtained from multiple sensors monitoring the phenomena or systems under consideration are analyzed to track their operating state or detect deviations from normal behavior. However, the effectiveness of these analysis algorithms depends on the quality of the input data, that is, their signal-to-noise power ratio. Since sensor data are typically subject to non-negligible measurement errors due to noise, the data recorded must be filtered to remove most of the noise while preserving the important waveform information. Though there exist a host of denoising algorithms, most are not inherently designed for real (or near-real) time data processing as they often lack an automated mechanism for selecting the best parameter values for denoising arbitrary measurement sequences using the prescribed filter.

We develop a simple and practical method for on-line denoising of sensor data streams with arbitrary waveform characteristics by using the Monte-Carlo *Stein's Unbiased Risk Estimate* (MC-SURE) algorithm [1]. This approach enables a blind optimization of the regularization parameters of a wide class of filters that seek to recover an arbitrary signal corrupted by additive white Gaussian noise (AWGN). The MC-SURE formulation is particularly suited for streaming data since it produces the optimal denoising parameter (defined using the mean-squared-error) for a chosen filter without any assumption about the underlying noise-free signal.

In this paper, we describe a strategy for adapting the MC-SURE algorithm to streaming data. First, in Section II, we describe the algorithm and discuss issues associated with its on-line implementation. Section III describes how we address these challenges. Results and discussions are given in Section IV and concluding remarks in Section V.

II. CONCEPTS AND CHALLENGES

The MC-SURE algorithm optimizes the (vector)-parameter λ of a continuous and (weakly) differentiable denoising function $f_\lambda(\cdot)$. Consider the noisy data

$$\mathbf{y} = \mathbf{s} + \mathbf{w} \tag{1}$$

comprised of a desired signal $\mathbf{s} \in \mathbb{R}^N$ and $\mathbf{w} \in \mathbb{R}^N$, a zero-mean AWGN with variance σ^2. The mapping

$$\hat{\mathbf{s}}_\lambda = f_\lambda(\mathbf{y}) \tag{2}$$

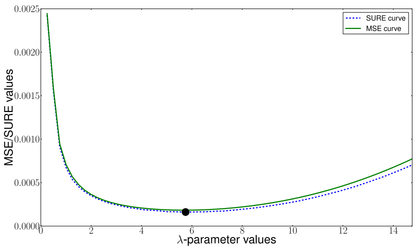
is the λ-parametrized data filtering operation that produces a signal estimate of $\mathbf{s}$. The MC-SURE procedure finds the optimal parameter for denoising $\mathbf{y}$ by minimizing a proxy of the mean-squared-error (MSE) criterion, namely, the *Stein's Unbiased Risk Estimate* (SURE) [2]. This SURE-statistic estimating the MSE is expressible [1] as

$$T_\lambda(\mathbf{y}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{s}}_\lambda\|^2 + \frac{2\sigma^2}{N}\mathrm{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\} - \sigma^2 \tag{3}$$

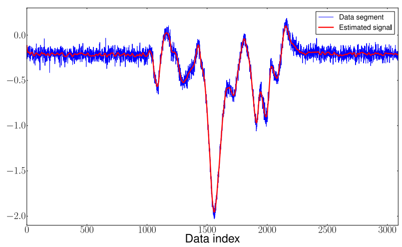
where $\mathrm{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\}$ denotes the divergence of the denoising function with respect to the data. The MSE-based optimal denoising parameter λ^* is thus the minimizer of the SURE-curve $\{T_\lambda(\mathbf{y}) : \lambda \in \mathbb{R}^K\}$. Fig. 1 illustrates the automated denoising of a data segment by a Gaussian filter whose parameter is identified using the SURE curve, which, as shown, closely approximates the MSE curve. For filters which do not have a closed-form expression for the divergence term in Equation 3, we can use a Monte-Carlo approach to estimate this term [1], thus providing us additional options in the selection of a noise-removal filter.

The MC-SURE procedure is particularly well-suited to on-line data denoising as it produces the *best* filter parameter without any assumption about the underlying signal. However, the application of the algorithm to streaming data presents some practical challenges. If we use a block-based approach, the need to minimize latency dictates that data blocks being processed must each have a limited number of observations. This introduces some performance issues as discussed in the following:

Noise estimation errors: The SURE-statistic formula from Equation 3 assumes the noise variance σ is known exactly.



(a) The SURE-curve approximates the MSE-curve and is thus used as a proxy for finding the parameter λ^* that minimizes the MSE.



(b) The data and the estimated signal obtained by filtering the data with the Gaussian filter with parameter $\lambda = \lambda^*$.

Fig. 1: Illustration of the MC-SURE procedure.

However, it has to be estimated in practice. In [1], Ramani *et al.* replace σ by its estimate computed using the Donoho median estimator method [3]. This approximates the noise standard deviation of a length-N dataset $\mathbf{y}$ by $\hat{\sigma} = M_{\mathbf{y}}/0.6745$ where $M_{\mathbf{y}}$ is the median of the $N/2$ wavelet coefficients at the finest scale. As the analysis in [1] was done on images with synthetically added AWGN, it is reasonable to substitute σ by $\hat{\sigma}$ because the noise is exactly white Gaussian and the number N of observations is very large (N $\geq 256^2$) However, in streaming/online data processing settings, the size of a data block will be relatively small. As a result, errors in estimating the noise term could be significant, leading to unreliable SURE-curves. For example, the length-N segments in Fig. 2 are from the same periodic signal and thus have the same frequency content in principle. When N is equal to 25, more substantial discrepancies are observed between SURE-curves as compared to the case where N equal 300. This can be attributed to the increased variance in estimating the noise as N gets smaller.

Errors in divergence-term computation: Given a length-N data segment $\mathbf{y}$ and a set of candidate denoising parameter values $\{\lambda_j : j = 1, 2, \ldots, J\}$, the reliability of the computed SURE-statistics $T_{\lambda_j}(\mathbf{y})$ also depends on the accuracy of the computed divergence term $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$. The divergence can be expressed as

$$\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\} = \lim_{\epsilon \to 0} E_{\mathbf{b}}\{\mathbf{b}^T(f_{\lambda_j}(\mathbf{y}+\mathbf{b}) - f_{\lambda_j}(\mathbf{y}))\}, \qquad (4)$$

where $\mathbf{b}$ is a zero-mean i.i.d. random vector with covariance $\epsilon^2\mathbf{I}$, and $E_{\mathbf{b}}\{\cdot\}$ denotes expectation with respect to $\mathbf{b}$. When a proper value of ϵ is chosen, $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$ is estimated using

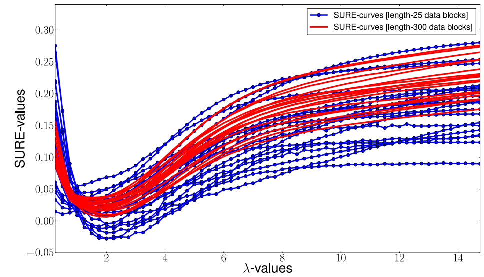


Fig. 2: SURE-curves for two groups of data segments: 20 length-25 data segments and 20 length-300 data segments

a Monte-Carlo (MC) approach: first, generate k realizations $\{\mathbf{b}_i\}_{i=1}^k$ of a length-N random vector $\mathbf{b}$ and then compute the k-MC-run divergence estimate

$$\widehat{\text{div}}_{\mathbf{y}}^{(k)}\{f_{\lambda_j}(\mathbf{y})\} = \frac{1}{\epsilon k}\sum_{i=1}^{k}\mathbf{b}_i^T(f_{\lambda_j}(\mathbf{y}+\mathbf{b}_i) - f_{\lambda_j}(\mathbf{y})), \qquad (5)$$

which amounts to averaging k single-MC-run divergence estimates. If N is large, a single MC-run suffices to generate a reliable estimate of divergence. However, as shown in Fig. 3, the estimation error increases rapidly with decreasing N. Thus, when N is relatively small, as would be the case in online block-based data processing, the estimates from a single run are unreliable as shown in Fig. 4. In such situations, an appropriate number, k, of MC-runs must be found on-the-fly to reliably compute the divergence-term for each data segment.

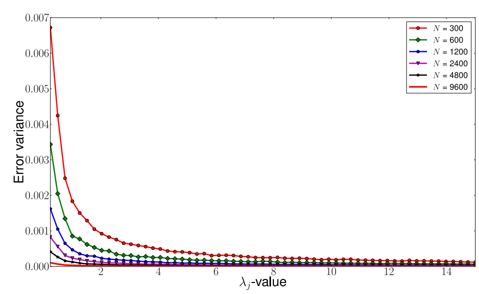


Fig. 3: Variance of the error in estimating the divergence-curve for different values of the data size N.

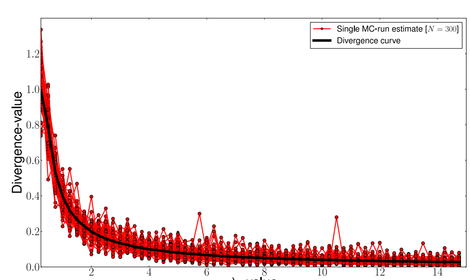
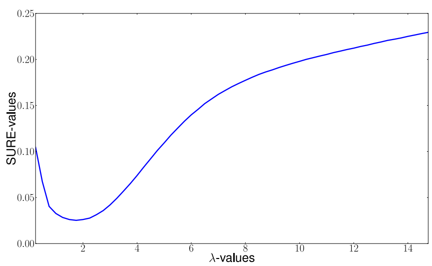
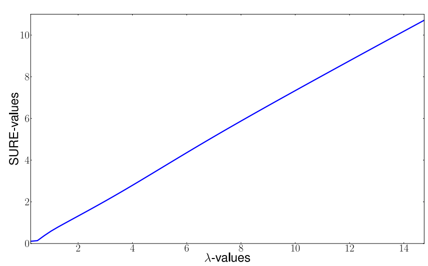


Fig. 4: Comparison of the divergence-curve with single-MC-run estimate when the data size is relatively low: N = 300 samples.

Detrimental effects of strong DC components: In addition to the effect of the small length of a data segment on the calculation of the noise and divergence estimates, our numerical experiments suggested that the MC-SURE procedure fails to return an appropriate denoising parameter for data with a strong DC-component. This issue is independent of the length of the data block under consideration. Fig. 5 illustrates the point. The data segments $\mathbf{z}$ and $\mathbf{z}_D$ are identical except for the presence of a strong DC-component: $\mathbf{z}_D = \mathbf{D} + \mathbf{z}$, where $\mathbf{D}$ is a constant vector. The application of the MC-SURE algorithm to $\mathbf{z}$ produces the SURE-curve in Fig. 5a, which reports the optimal filter parameter value of $\lambda^* = 2$. We expect that a similar denoising parameter value should be necessary for $\mathbf{z}_D$, but the corresponding SURE-curve, shown in Fig. 5b, erroneously reports that no denoising is needed ($\lambda^* = 0$). In general, when the data under consideration have a very strong DC component, the MC-SURE algorithm appears to always report that no denoising is needed, regardless of the amount of noise present.



(a) SURE-curve of $\mathbf{z}$



(b) SURE-curve of $\mathbf{z}_D$ ($\mathbf{z}$ plus DC-component)

Fig. 5: Comparison of the SURE-curves generated using the datasets $\mathbf{z}$ and $\mathbf{z}_D$. Although the two SURE-curves differ significantly, $\mathbf{z}$ and $\mathbf{z}_D$ only differ by a DC-component term.

III. A Practical Strategy for Denoising Sensor Data Streams

Consider a sensor data stream $\mathbf{y} = \{y_n : n = 0, 1, \dots\}$ comprised of a desired signal $\mathbf{s} = \{s_n : n = 0, 1, \dots\}$ plus AWGN. The model assumes that the noise parameter varies as time elapses, albeit very slowly. Samples from $\mathbf{y}$ are taken in as successive data blocks of sizes which are appropriately chosen. The i-th data block is denoted by $\mathbf{y}_i = \{y_n : N_i \leq n < M_i\}$, where $N_{i+1} = M_i - L$ and L is the overlap between consecutive data blocks. The stream of data blocks $\{\mathbf{y}_i : i = 0, 1, \dots\}$ is denoised one at a time. The noise in the i-th data block $\mathbf{y}_i$ is filtered out using the prescribed filter $f_\lambda(\cdot)$ with parameter λ set to λ_j^*, the best MC-SURE-based parameter value. The i-th data block output, $\hat{\mathbf{s}}_i = f_{\lambda_j^*}(\mathbf{y}_i)$, is the sub-signal of $\mathbf{s}$ contained in $\mathbf{y}_i$. The estimate of the portion of $\mathbf{s}$ that is available after processing the ($\mathcal{I}$+1)-th data block $\mathbf{y}_{\mathcal{I}+1}$ is denoted by $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$. This current signal estimate $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$ is formed by stitching together the available sub-signal estimates: $\{\hat{\mathbf{s}}_0, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_{\mathcal{I}+1}\}$. This is done recursively via

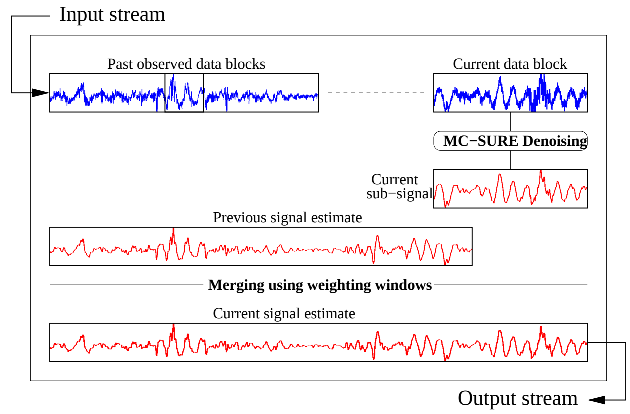


Fig. 6: Schematic representation of the block-based, on-line, data de-noising framework using the MC-SURE algorithm.

$$\mathbf{s}^{(\mathcal{I}+1)} = \mathbf{s}^{(\mathcal{I})} \odot \mathbf{W}^{(\mathcal{I})} + \hat{\mathbf{s}}_{\mathcal{I}+1} \odot \mathbf{W}_{\mathcal{I}+1}, \tag{6}$$

where $\mathbf{s}^{(\mathcal{I}+1)}$ is new signal-estimate constructed by properly merging via tapered windows the newly obtained sub-signal estimate $\hat{\mathbf{s}}_{\mathcal{I}+1}$ with the previous signal-estimate $\mathbf{s}^{(\mathcal{I})}$. The operator $\odot$ denotes element-by-element multiplication. The terms $\mathbf{W}^{(\mathcal{I})}$ and $\mathbf{W}_{\mathcal{I}+1}$ denote here one-sided cosine tapering windows although other types of tapering windows are also an option. The recursive procedure is initialized with $\mathbf{s}^{(0)} = \hat{\mathbf{s}}_0$. This data processing strategy is illustrated in Fig. 6. For this system to be effective in denoising streaming data, individual data blocks must be properly denoised using the MC-SURE in spite of their relatively short lengths or the presence of DC terms. We address the issues identified in Section II as follows:

Removal of DC components: To suppress the detrimental effects of DC components on the performance of the MC-SURE algorithm, we first pass each data block $\mathbf{y}_i$ through a notch-filter centered at the DC-frequency. A data block without a DC-component will be indifferent to this pre-filtering. For simplicity, we may assume hereon that each data block is free of a DC-component as it would have been removed at this stage, if present.

Noise estimation: We previously noted that estimating the noise term σ using only the limited number of samples from a data block $\mathbf{y}_i$ produces an inaccurate estimate which deteriorates the accuracy of the computed SURE-curve. We mitigate the problem by combining observations from a chosen number of consecutive data blocks to generate a better estimate of σ. Since noise characteristics are expected to change slowly with time, it is reasonable to assume that the noise observations from M consecutive data blocks come from i.i.d. Gaussian distributions with parameter σ. The Donoho median estimator produces a separate estimate of σ from each of the M previous data blocks. When processing the current data block $\mathbf{y}_{\mathcal{I}}$, the noise estimates from the M previous data blocks $\{\hat{\sigma}_{\mathcal{I}-M+1}, \hat{\sigma}_{\mathcal{I}-M+2}, \ldots, \hat{\sigma}_{\mathcal{I}}\}$ are combined to produce a more reliable estimate of σ via

$$\bar{\sigma}_{\mathcal{I}} = \frac{\sum_{i=\mathcal{I}}^{\mathcal{I}-M+1} N_i \hat{\sigma}_i}{\sum_{i=\mathcal{I}}^{\mathcal{I}-M+1} N_i} \tag{7}$$

where N_i denotes the length of the i-th data block. The weighted averaging takes into consideration the unequal reliabilities of individual noise estimates and gives more weight to those from longer data blocks as they are expected to be more reliable.

Divergence-curve computation: For the grid of parameter-values $\{\lambda_j : j = 1, 2, \ldots, J\}$, the divergence-curve estimated via Equation 5 using k MC runs is denoted by

$$d_j^{(k)} = \widehat{\mathrm{div}}_{\mathbf{y}_{\mathcal{I}}}^{(k)} \{f_{\lambda_j}(\mathbf{y}_{\mathcal{I}})\}, \; j = 1, 2, \ldots, J \tag{8}$$

For a fixed j, $\{d_j^{(k)} : k = 1, 2, \ldots\}$ is a converging sequence of estimates of $\mathrm{div}_{\mathbf{y}}\{f_{\lambda_j}\}$. The j-th element of the divergence-curve is thus estimated by $d_j^{(K_j)}$ with the number of MC-runs K_j chosen such that

$$r_j^{(K_j)} = \frac{|d_j^{(K_j+1)} - d_j^{(K_j)}|}{|d_j^{(2)} - d_j^{(1)}|} < \delta, \tag{9}$$

where δ is the convergence stopping criterion. Seeking K_j for each of the J divergence-values may be impractical: The convergence rate of $\{r_j^{(k)} : k = 1, 2, \ldots\}$ differs from one j-index to another. Thus, instances might occur where, for one or a few j-indexes, an unreasonably large number of runs is needed to satisfy Equation 9. A group convergence requirement is then used to avoid such situations: for all parameter values λ_j, the divergence-value is estimated by $d_j^{(K)}$ with the number of MC-runs chosen K such that

$$r^{(K)} = \frac{1}{J} \sum_{j=1}^{J} r_j^{(K)} < \delta. \tag{10}$$

The k-indexed sequence $r^{(k)}$ goes from one to zero as the divergence-curve estimate $\{d_j^{(k)} : j = 1, 2, \ldots, J\}$ converges with additional MC runs. Fig. 7 shows an example profile for $r^{(k)}$, where the choice of δ = 0.02 leads to the termination of the divergence-curve estimation after K = 67 MC simulations.

Online selection of data block size: To avoid potential spatial aliasing issues in the processing of the streaming data segments, we select the sizes of data blocks based on waveform structure information computed from the data. We use techniques from scale-space theory [7] to identify the typical length (e.g., scale) of the most salient signal structures in a dataset. By choosing the size of the next block sufficiently larger than the data scale, we ensure that enough samples are taken to avoid potential spatial aliasing issues. Since processing latency has to remain minimal, the

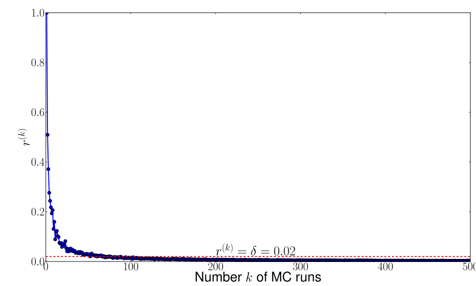


Fig. 7: Illustration of the profile of the sequence $r^{(k)}$,q which is used to track the convergence of the divergence-curve estimate.

sizes of windows must be bounded above. At the $\mathcal{I}$-th data block, the computed data scales for the last P data blocks are $\{\Delta_{\mathcal{I}-P+1}, \Delta_{\mathcal{I}-P+2}, \ldots, \Delta_{\mathcal{I}}\}$. The size $N_{\mathcal{I}+1}$ of the ($\mathcal{I}$+1)-th data block is determined via

$$N_{\mathcal{I}+1} = \min\left\{N_{\max}, \theta \cdot \max\left\{\{\Delta_i\}_{i=\mathcal{I}-P+1}^{\mathcal{I}}\right\}\right\} \tag{11}$$

where θ is a multiplicative factor and $N_{\max}$ is the maximal data block size.

IV. Results and Discussions

We tested the denoising strategy using both synthetic and real datasets. Sample results and implementation steps are illustrated here with a real-world dataset: a sequence of vehicle body acceleration response measurements. Table I gives the parameter values used in the implementation. We observed (as noted in [1]) that MC-SURE is very robust to variations in the value of ϵ, which can change from 1 to 10^{-12} without discernible changes in performance. Since no previous data are available to systematically decide the initial data block size, it is chosen equal to $2N_{\max}$. The sizes of subsequent blocks are computed via Equation 11 using the largest of the scales of the two blocks preceding them (i.e., $P = 2$).

TABLE I: Values used for the algorithm parameters.

Parameter	Chosen value
Standard deviation of probing noise: ϵ	0.0002
Maximum allowed segment size: $N_{\max}$	500
Fixed overlap between segments: L	30
Multiplicative factor: θ	24
Number P of past blocks	2
Number M of past blocks	5

Table II gives the noise statistics and denoising parameters computed for 20 consecutive data blocks. The parameter $\bar{\sigma}_i$, used as noise term for the i-th data block, is the weighted average of the individual block noise estimates from the previous five data blocks (i.e., $M = 5$). The denoising parameter λ_i^* is the standard deviation of the Gaussian filter as determined by the MC-SURE algorithm. Sample results from our denoising strategy are shown in Fig. 8. These results support the premise that it is possible to automatically select the parameters for a denoising algorithm to process streaming data.

TABLE II: Computed statistics/parameters for 20 segments.

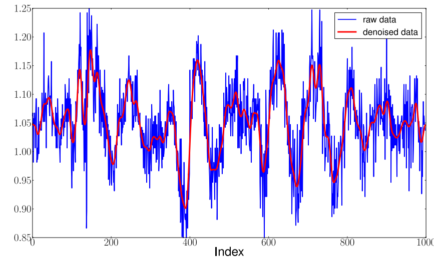
Data block	Scale	Length	$\hat{\sigma}_i$	$\bar{\sigma}_i$	λ_i^*
1	12	1000	0.0396	0.0396	3.75
2	12	244	0.0553	0.0427	4.25
3	15	244	0.0480	0.0435	7.0
4	19	360	0.0357	0.0420	5.0
5	13	456	0.0398	0.0416	3.25
6	13	456	0.0464	0.0440	4.25
7	11	312	0.0405	0.0419	3.25
8	15	312	0.0527	0.0429	3.25
9	12	360	0.0476	0.0451	3.75
10	13	360	0.0381	0.0451	3.25
11	31	312	0.0323	0.0423	3.75
12	13	500	0.0368	0.0411	2.75
13	14	500	0.0322	0.0372	3.0
14	13	336	0.0382	0.0355	3.5
15	13	336	0.0393	0.0357	3.5
16	15	312	0.0381	0.0366	1.0
17	13	360	0.0486	0.0388	0.75
18	11	360	0.0590	0.0450	0.5
19	12	312	0.0554	0.0483	3.75
20	14	288	0.0650	0.0533	0.5

V. Conclusion

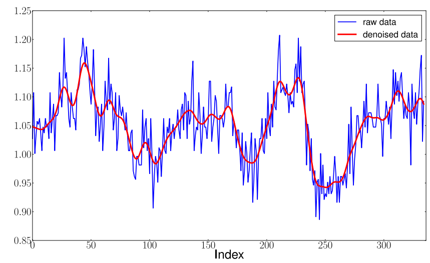
We proposed a practical strategy to automatically denoise streaming data that have been corrupted by (approximately) Gaussian noise. First, we recognized that the MC-SURE algorithm is well-suited to denoise sensor data streams since it prescribes a procedure for optimizing the regularization parameters of a wide class of denoising filters without any assumption about the underlying signal of interest. Second, we identified and addressed the challenges in extending the MC-SURE algorithm to streaming data. Results from our tests on real datasets that reasonably fit the data model show good denoising performance. Future research directions would include a comparison of our approach with other methods for denoising continuous data, such as wavelets [8], as well as a sample-based approach to further reduce the latency.

Acknowledgment

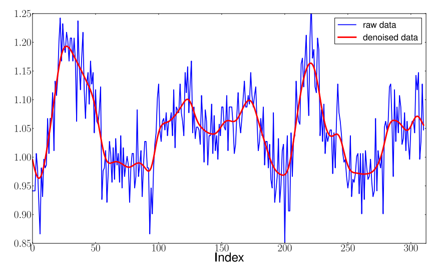
LLNL-CONF-569332. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.



(a) Automated (Gaussian) denoising: 1-st segment



(b) Automated (Gaussian) denoising: 14-th segment



(c) Automated (Gaussian) denoising: 7-th segment

Fig. 8: Illustration of denoising with the Gaussian filter.

References

[1] S. Ramani, T. Blu and M. Unser, *Monte-Carlo SURE: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms*, IEEE Transactions of Image Processing, Vol. 17, No. 19, September 2008.

[2] C. Stein, *Estimation of the Mean of a Multivariate Normal Distribution*, Annals of Statistics, Vol. 9, Pages 1135-1151, 1981.

[3] D. L. Donoho, *Adapting to Unknown Smoothness via Wavelet Shrinkage*, J. Amer. Statist. Assoc., Vol. 90, No. 432, Pages 1200-1224, 1995.

[4] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications. 4th Edition*, Prentice Hall, 2007.

[5] S. Orfanidis, *Introduction to Signal Processing*, Prentice Hall, 1995.

[6] A. P. Witkin, *Scale-space Filtering*, Proc. 8th Int. Joint Art. Intell., Karlsruhe, Germany, Pages 1019-1022, 1983.

[7] T. Lindeberg, *Scale Space Theory: A Basic Tool for Analyzing Structures at Different Scales*, Journal of Applied Statistics, Vol. 21, No. 2, Pages 224-270.

[8] H. de Oliveira Mota, F. H. Vasconcelos, and R. M. da Silva, *A Real-Time System for Denoising of Signals in Continuous Streams Through the Wavelet Transform*, IEEE International Symposium on Signals, Circuits and Systems, (ISSCS), Volume 2, Pages 429-432, 2005.